# EXHIBIT D

TEXAS INSTRUMENTS

*Application Report*
*SLUA359 – November 2005*

# *Using SHA-1 in bq20zxx Family of Gas Gauges*

*Travis Neely* *PMP Portable Power*

**ABSTRACT**

The bq20zxx Impedance Track™ family of gas gauge ICs includes a highly sophisticated authentication algorithm, known as SHA-1, which requires little setup and development time and provides an effective, secure battery design.

## 1 Introduction

Battery counterfeiting is a major problem confronting original equipment manufacturers (OEM) today. One of the most effective methods to counter this issue is with the use of SHA-1 authentication routines in battery designs. Using this approach ensures that the OEM can track the suppliers for battery replacements. With this anti-counterfeiting algorithm, only battery packs manufactured by authorized subcontractors using the bq20zxx Impedance Track™ gas gauge IC with the SHA-1 can be integrated into OEM system designs. The SHA-1 authentication key in the bq20zxx can be regulated and tracked by the OEM. Multiple subcontractors can be supplied with different authentication keys for even greater security and regulation

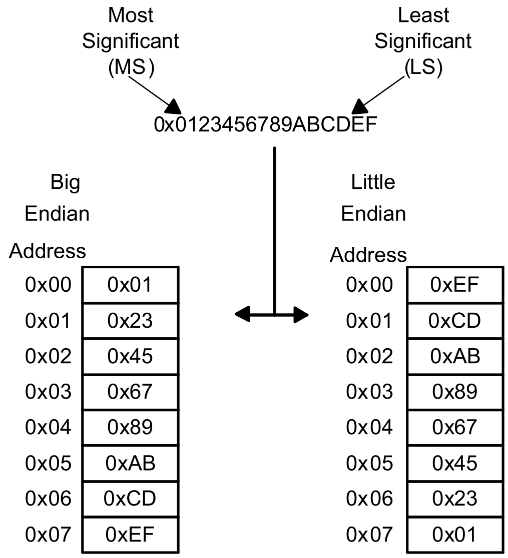
## 2 Explaining Little Endian

The SHA-1 features of the bq20zxx use SMBus string reads and writes. Because SMBus communications is based on the Little Endian scheme of byte ordering, an understanding of Little Endian can reduce the complexity of the development process.

Two, byte-ordering schemes are used when storing multibyte data in memory: Little Endian and Big Endian. In Big Endian ordering, the most significant byte (MSB) is stored at the lowest possible memory address. This ordering method is most routinely used in Motorola processors.

In Little Endian ordering, the least significant byte (LSB) is stored first in memory at the lowest address. Little Endian is used in Intel processors. It is also used by the SMBus data transfer method for multibyte data transfers. In other words, SMBus always transfers the LSB first.

Impedance Track is a trademark of Texas Instruments.




## *2.1 What is SHA-1?*

This document describes the SHA-1 functions only as they are used with the bq20zxx gas gauge IC. More complete explanations of the SHA-1 algorithm are available in many articles and books. For example, see www.faqs.org/rfcs/rfc3174.html for an excellent description of the SHA-1 algorithm and some C-code examples of how to implement it.

Three primary bq20zxx functions are used to implement the SHA-1 security feature in a system design. The first function is the SHA-1 challenge. The challenge is a 20-byte (160-bit ) string sent to the bq20zxx by the host. The SHA-1 algorithm in the bq20zxx then is required to send back a response. The 20-byte challenge, located at SMBus command 0x2f is a 20-byte SMBus string write. As with the rest of the SHA-1 function, this challenge string is Little Endian.

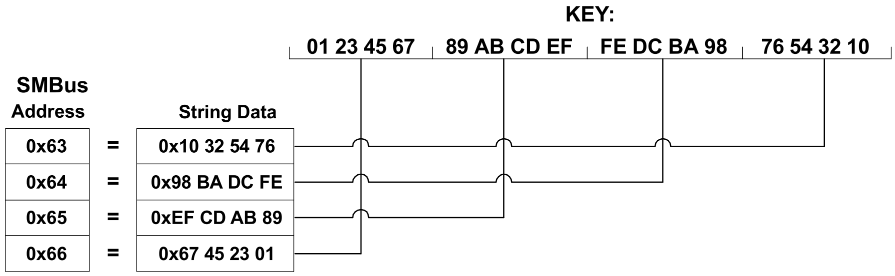
The second bq20zxx function used to implement the SHA-1 feature is the SHA-1 response. The response is a 20-byte string read. Once a challenge is given and the bq20zxx is given time to compute the response, it is available in the same SMBus command as the challenge (command 0x2f).

The third function is the SHA-1 authentication key. The authentication key is the primary function of the SHA-1 algorithm. The key is input during production only by using the *Gold Data Flash File* methodology explained in a later section entitled *How to Set Up for Production*. Once the key is written and the part sealed, it is completely inaccessible. It must be kept secret. With the key unknown, it is virtually impossible for the challenge/response pattern to be decoded. If the key is compromised, the authentication is no longer effective.

## *2.2 How to Use SHA-1 in the bq20zxx*

The following two steps are used to implement the SHA-1 algorithm in the bq20zxx.

1. Create a unique authentication key, and write it to the part during assembly: The authentication key resides in the SMBus addresses 0x63-0x66 in 4-byte strings. The four strings are read/write accessible until the bq20zxx is sealed. When written using an SMBus string write command, they are retained permanently in flash memory and can only be changed when the bq20zxx is unsealed. They are stored in Little Endian format as shown in the following diagram. The SHA-1 authentication key defaults to *0123456789abcdeffedcba9876543210* in the bq20zxx. This is a default and is not intended for production. It should be changed to a unique key prior to production to ensure that security is not compromised.




2. Implement SHA-1 in the OEM host system.
   a. The host has to know the SHA-1 authentication key:
      The host must know the key defined in step 1. This key is used in the host system to determine what the response should be.
   b. The host has to issue a random challenge:
      The host sends a challenge using a 20-byte string write to SMBus command 0x2f in Little Endian format. It is important that the challenge be random every time to ensure security. Here is an example of a challenge and writing it in Little Endian:

      Using the example of:

      | |
      |---|
      | **0x20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 3E 2F 30 31 32 33** |

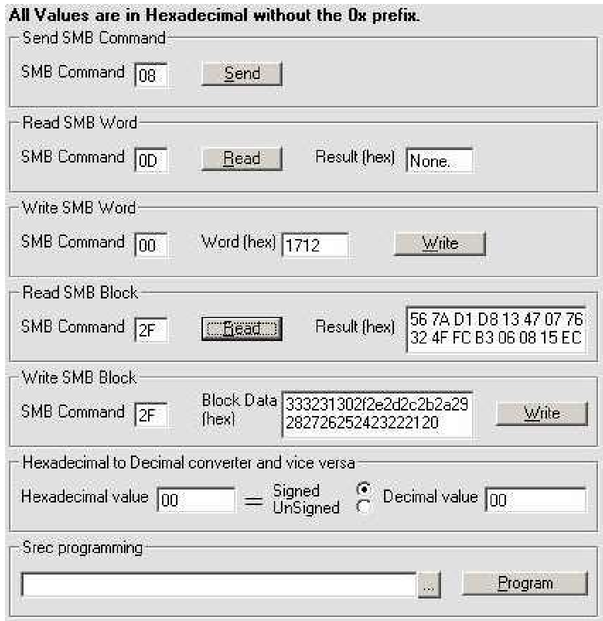
      Must be written in Little Endian as follows:

      | |
      |---|
      | **0x33 32 31 30 2F 2E 2D 2C 2B 2A 29 28 27 26 25 24 23 22 21 20** |

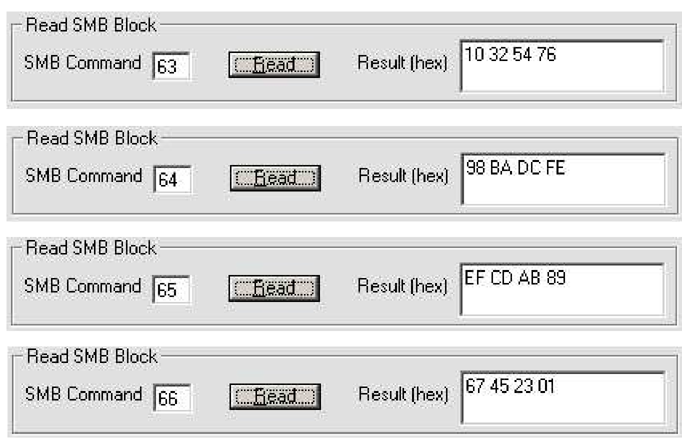
   c. The host computes the response:
      With the known SHA-1 authentication key and random challenge, the host computes the anticipated response from the bq20zxx.
   d. Bq20zxx computes the response:
      The bq20zxx computes the response at the same time that the host is computing it. The bq20zxx should be given greater than 100 ms to compute the response and put it into memory for retrieval.
   e. The host has to read the response:
      The host reads a response from the same command (0x2f) to which the challenge was written. The response is a 20-byte string read in Little Endian format.
   f. The host must validate the response:
      The host must compare the response read from command 0x2f in the bq20zxx to what was computed in step *2c*.
   g. If the response is validated, then the battery is authorized. Otherwise, the host can reject the battery pack.

## *2.3 Experimenting With the bq20zxx Evaluation Software:*

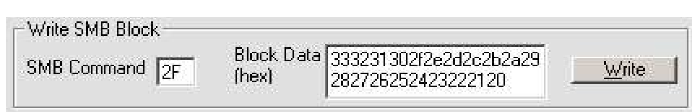
The PRO screen of the bq20zxx Evaluation software can be used to experiment with the SHA-1 features. This screen includes the SMBus read and write block functions that are required for SHA-1. Use the procedure described in the preceding section entitled *How to Use SHA-1 in the bq20zxx* with the following functions:




1. Use the Read SMB Block frame to verify that the key is written to the desired value using the four reads to SMBus commands 0x63-66. They should read as follows by default. Notice that they report in Little Endian.




2. If the authentication key needs to be changed, it can be modified with an SMBus write to the same SMBus commands 0x63-66 in Little Endian.
3. Now, write the challenge in Little Endian to the bq20zxx using the Write SMB Block Frame:



4. Then, use the Read SMB Block Frame to read the Response from the bq20zxx:


5. Notice that only 16 bytes fit in the Read SMB Block Frame. The entire response is in the Result window but not wholly visible. Select and highlight the Result data and paste it into a text editor to see the entire result as follows:

**56 7A D1 D8 13 47 07 76 32 4F FC B3 06 08 15 EC 23 5C AB FE**

## 2.4 How to Set Up for Production:

Setting up the SHA-1 for production assembly is simple using the BqTester *Gold Data Flash File* methodology for testing production modules. See *Using the bqTester Software* (SLUA352) application report or *bqMulti-Site Tester* user guide at www.ti.com for more information.

Using the bq20zxx EV software, set up the module as required for the application. Many of the TI documents in the corresponding bq20zxx IC product folder at www.ti.com can assist the user in setting up the module. As an example, for the bq20z80, these include:

- *Preparing Optimized Default Flash Constants For Specific Battery Types* application report (SLUA334)
- *bq20z80 EVM Data Flash Settings for Number of Serial Cells and Pack Capacity* application report (SLVA208)
- *Pack Assembly and the bq20z80* application report (SLUA335)
- *Exploring the bq20z80 Impedance Track Evaluation Kit* application report (SLUA351)
- *Pack Assembly and the bq20z80* application report (SLUA335)

Once a module is configured as required for the particular application, the desired SHA-1 authentication key needs to be saved into the bq20zxx module as explained in the previous section *How to Use SHA-1 in the bq20zxx*. With this completed, the SHA-1 key is stored, and the data flash meets the requirements for the *Gold Data Flash File*.

As explained in *Using the bqTester Software* application report or the *bqMulti-Site Tester* user guide, create the *Gold Data Flash File*. Use this file with bqTester or bqMulti-Site Tester Software for module testing in production. Be sure to seal the bq20zxx as explained in the corresponding bq20zxx data sheet. All modules produced with this *Gold Data Flash File* and bqTester or bqMulti-Site Tester will have the same hidden SHA-1 key and be ready for use.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| **Products** | | **Applications** | |
|---|---|---|---|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DSP | dsp.ti.com | Broadband | www.ti.com/broadband |
| Interface | interface.ti.com | Digital Control | www.ti.com/digitalcontrol |
| Logic | logic.ti.com | Military | www.ti.com/military |
| Power Mgmt | power.ti.com | Optical Networking | www.ti.com/opticalnetwork |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| | | Telephony | www.ti.com/telephony |
| | | Video & Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless |

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265